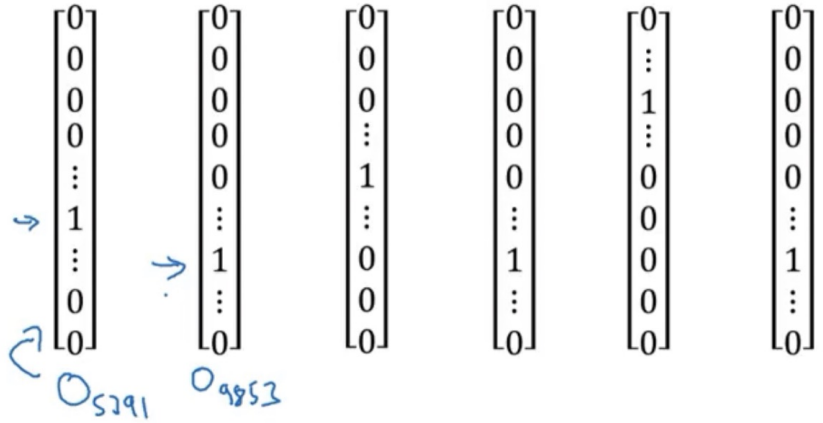
Word Representation

1-hot representation:

V = [a, aaron, man, Woman, King, Queen, Apple, Orange ....]

Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)



Problem: Correlations between any 2 words are 0

In other word, Networks using this type of word embeddings don't understand meanings and relations of words.

e.g. I want a glass of orange juice _____

I want a glass of apple ?

Lack of understanding of similarities between words.

Featurized representation: word embedding

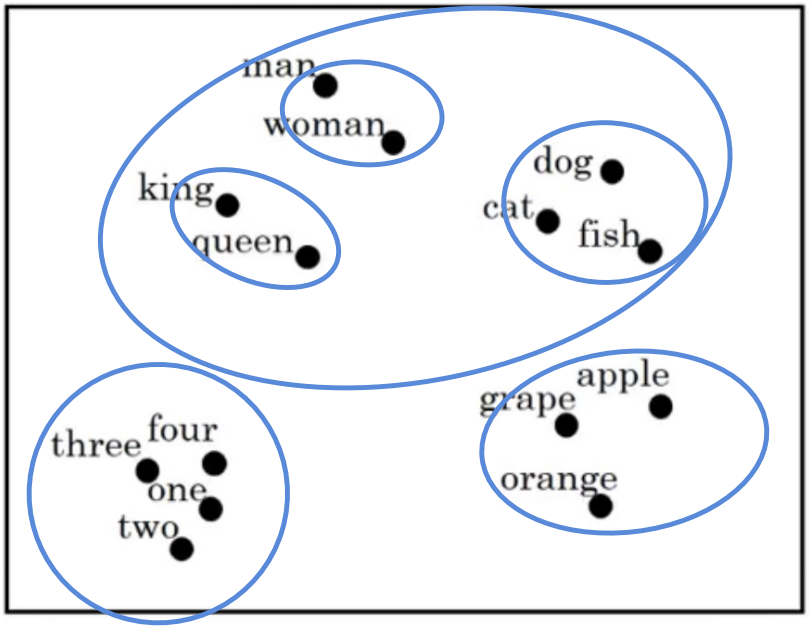
	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
...	...	...				
size						
cost						
alive						
verb						

300 300D vector

e_{5391} e_{9853}

Andrew Ng

Visualizing word embeddings



t-SNE =
map 300D feature vector to 2D space

- preserving spacial relationships
- Not a linear mapping
  -> lengths are distorted

Transfer Learning and word embeddings

Common Practice:

1) Learn word embeddings from large corpus (1-100B words)
   (or download pre-trained embedding online)
2) Transfer embedding to new task with smaller training set (say, 100k words)
3) (optional) Continue to finetune the word embeddings with new data.

Word Embedding in NLP is similar to face/image encoding in facial recongnition task.

Properties of Word Embeddings

motivation problem:

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	−1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

e_{man} e_{woman} e_{king} e_{queen}

Given the above word embeddings, how can computers solve the problem below?

Man ⟶ Woman as King ⟶ (?)

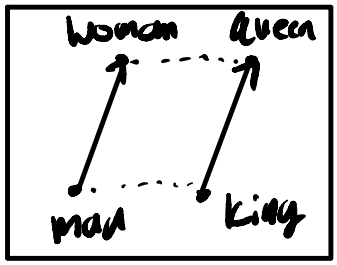
One possible aproach:

$$e_{man} - e_{woman} \approx \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{king} - e_{queen} \approx \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow e_{man} - e_{woman} \approx e_{king} - e_? \quad e_? \approx e_{queen}$$

Analogy using word vectors:



300D

$$e_{man} - e_{woman} = e_{king} - e_? \quad e_? \approx e_w$$

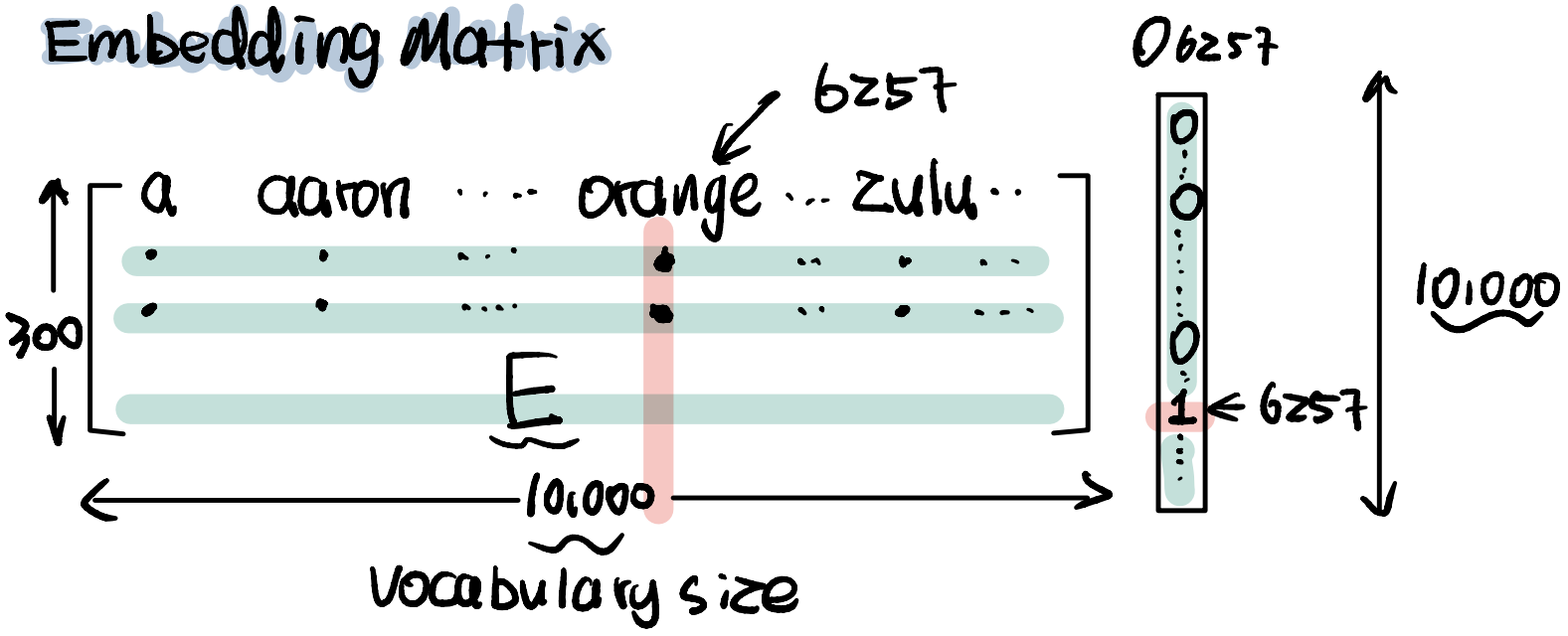
Find word w:

$$\underset{w}{\text{argmax}}\ sim(e_w, e_{king} - e_{man} + e_{woman})$$

$$sim(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

Learning Word Embeddings

Embedding Matrix



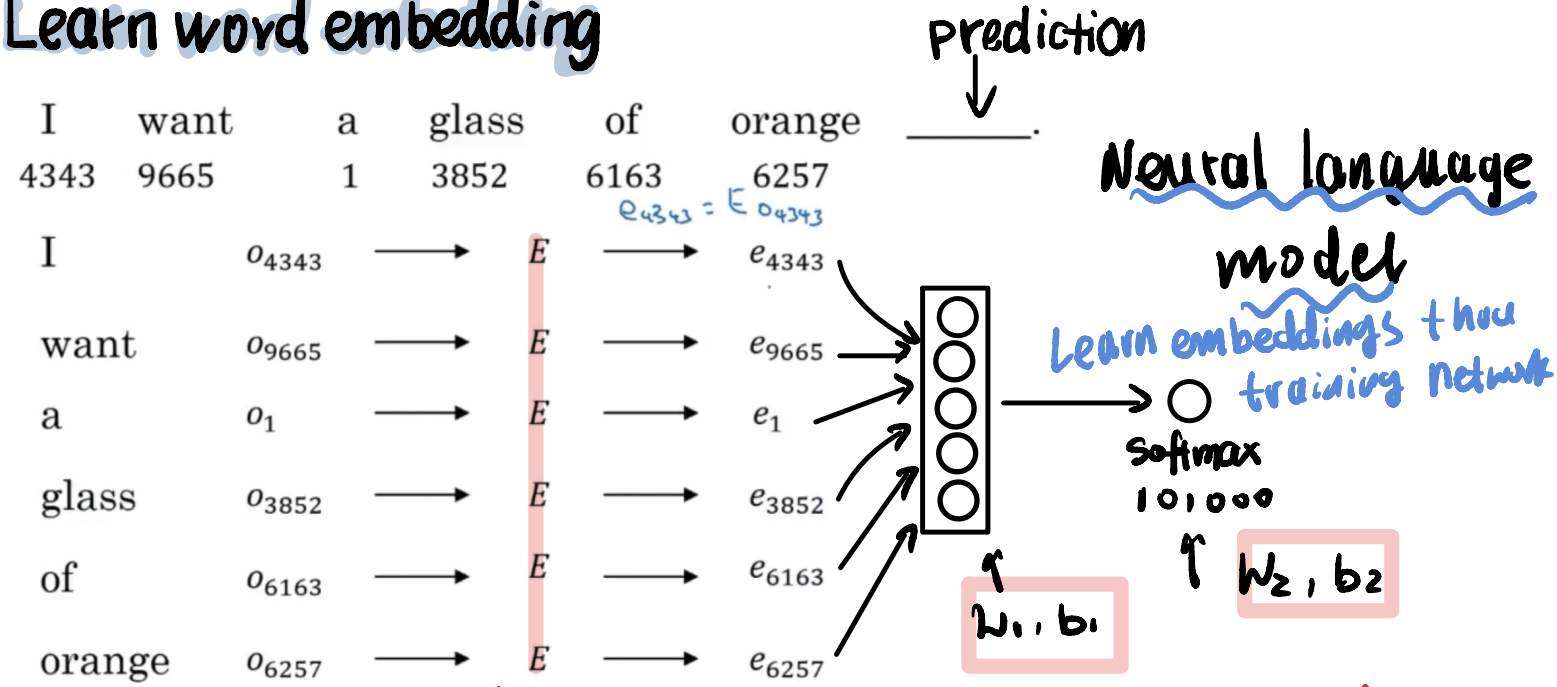
$E \cdot O_{6257} = e_{6257}$ · shape = (300, 1)

→ embedding for O_{6257} (orange)

$E \cdot O_j = e_j$

= embedding for word j

Learn word embedding



Other context/target pairs

I want a glass of orange juice to go along with cereal

context target

Possible context:

- Last 4 word
- Last 1 word
- 4 words on left and right
- Nearby 1 word → skip gram.

...

word2vec

I want a glass of orange juice to go along with cereal

skip gram:

context:	Target:	
orange	juice	a random chosen word with a window close to context
orange	glass	
orange	my	

..

As we can guess, the training data generated in this way won't do well on the supervised learning task which maps context to target. However, it's proven to be very effective for the network to learn the embedding matrix.

model:

Vocab size = 10,000k

$X \longrightarrow y$

context c ("orange") $\longrightarrow$ Target t ("juice")

$O_c \longrightarrow E \longrightarrow e_c \longrightarrow O_{softmax} \longrightarrow \hat{y}$

$e_c = E O_c$

softmax: $p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$

θ_t: Param associated with output t

$L(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$

$y = \begin{bmatrix} 0 \\ \vdots \\ i \\ \vdots \\ 0 \end{bmatrix}$ 10,000

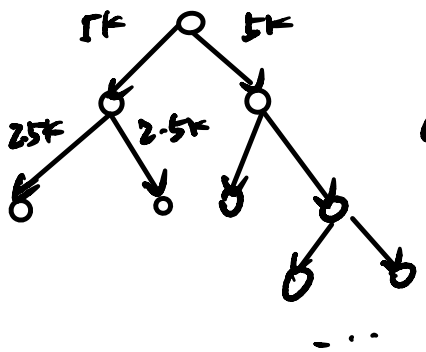
Problems with softmax classification

softmax: $p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$

This computation of this summation over all vocabulary is very expensive

One direct solution:

Hierchical softmax



Won't be a balanced binary tree: less frequent words will be berrid deeper down the tree.

*$p(c)$: choosing context word won't happen within a uniform distribution. Avoid drawing too many common words such as the, of, a, ...

Negative Sampling : Another algorithm that avoids the Softmax problem.

I want a glass of orange juice to go along with cereal

context	word	target
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

Randomly picked from vocab, neg examples (K-neg-ex)

Forming A Supervised learning task.

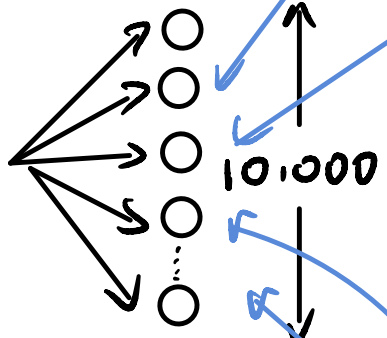
c → x ← t

y

$$P(y=1 \mid c, t) = \sigma(\theta_t^T e_c)$$

Orange
6257

$O_{6257} \longrightarrow E \longrightarrow e_{6257}$



for juice ?

for book ?

for King ?

for ...

10,000 binary classification problem.

Instead of calculating Softmax over entire vocabulary set, we only perform k+1 binary classification problems. (K neg + 1 pos example).

Selecting words to generate negative examples:

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10000} f(w_j)^{3/4}}$$

Glove word vectors

I want a glass of orange juice to go along with cereal

c, t

X_{ij} = # times j appears in context of i

X_{ij} = X_{ji}, depending on how "x appears in context of y" is defined.

Model:

$$\text{minimize} \sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij}) \left(\theta_i^T e_j + b_i + b_j - \log X_{ij}\right)^2$$

"$\theta_t^T e_c$" ≈ correlation ≈ # times X_{ij}

weighting term:

$f(X_{ij}) = 0$ if $X_{ij} = 0 \rightarrow 0 \log 0 = 0$

θ_i, e_j are symmetric

destressing → this, is, a, an ...

adressing → durian ...

(final) $e_w = \frac{e_w + \theta_w}{2}$

* Featurization might not be interpretable.

$\rightarrow (A\theta_i)^T (A^{-T} e_j) = \theta_i^T A^T A^{-T} e_j = \theta_i^T e_j$

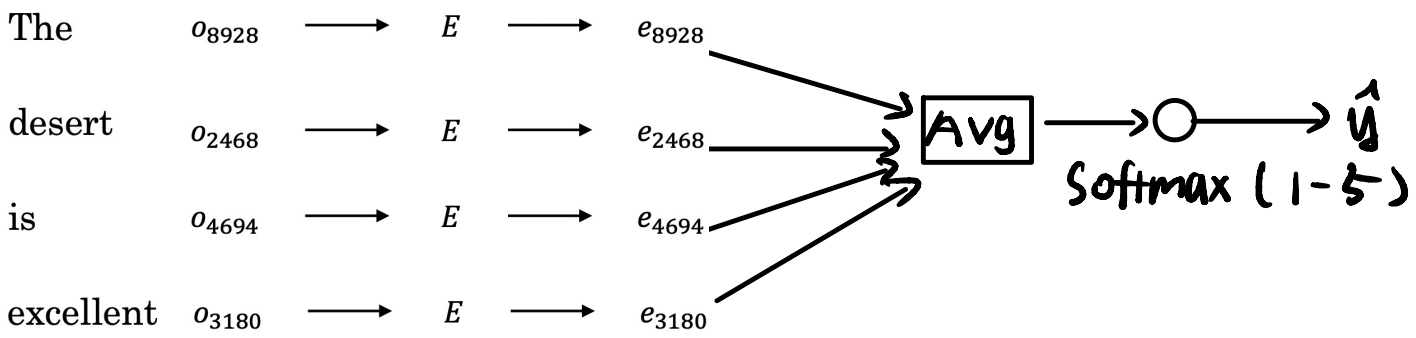
Linear transformations may shift featurization away from our natrual interpretation.

Sentiment Classification:

mapping from x ⟶ y

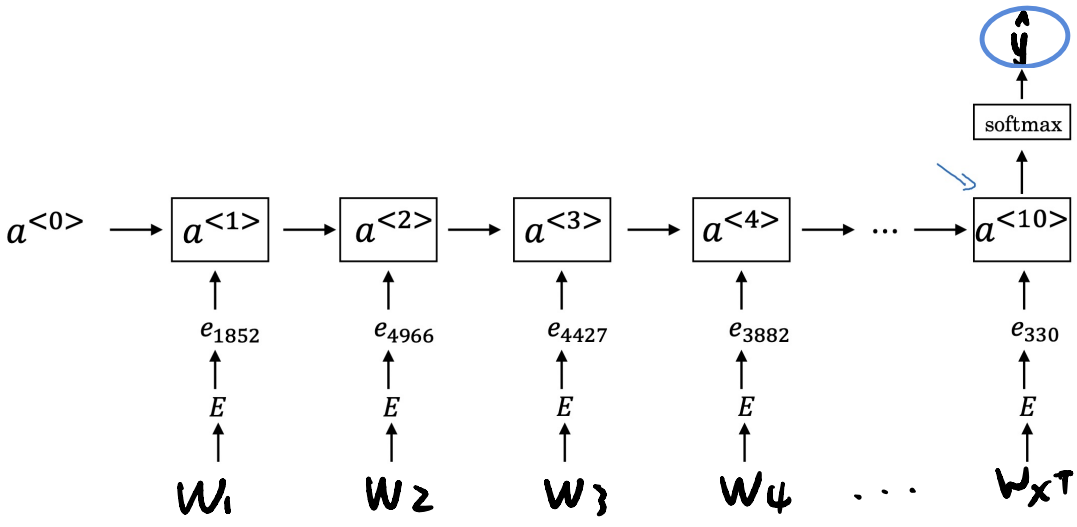
The dessert is excellent.	★★★★☆
Service was quite slow.	★★☆☆☆
Good for a quick meal, but nothing special.	★★★☆☆
Completely lacking in good taste, good service, and good ambience.	★☆☆☆☆

Simple sentiment classification model:



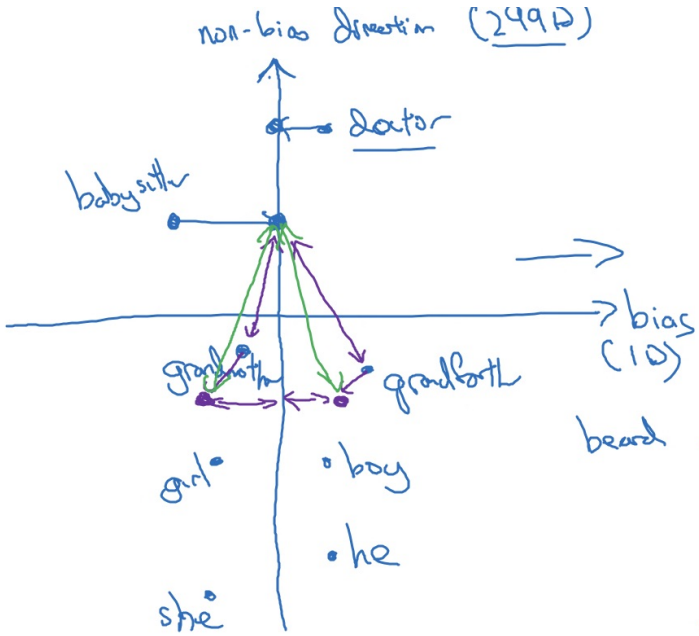
This simple model performs very poorly when input sentence contains more complex logic and sentiments e.g. "Completely lacking in good taste and good service"

Solution: RNN for Sentiment classification



many-to-one RNN

Debiasing word embeddings



1. Identify bias direction.

$e_{he} - e_{she}$

$e_{male} - e_{female}$

$\vdots$

average

2. Neutralize: For every word that is not definitional, project to get rid of bias.

3. Equalize pairs.

grandmother — grandfather

girl — boy

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Andrew Ng